Introduction to Artificial Intelligence (AI)

Computer Science cpsc502, Lecture 10

Oct, 13, 2011

Today Oct 13

- Inference in HMMs
- More on Robot Localization

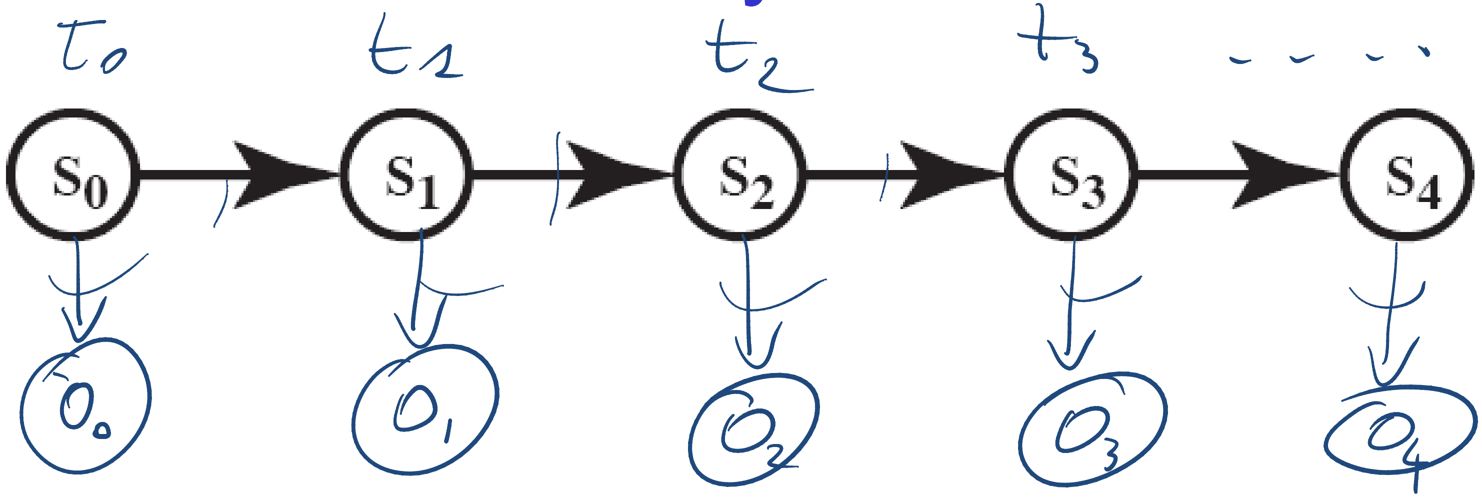
R&Rsys we'll cover in this course

Problem		Environment: Deterministic	Environment: Stochastic
Static	Constraint Satisfaction	*Vars + Constraints* Arc Consistency SLS Search	
Static	Query	*Logics* → Propositional → First Order → . . . . Search	*Belief Nets* Var. Elimination Approx. Inference Temporal. Inference
Sequential	Planning	*STRIPS* actions precs effects Search	aka influence diagrams *Decision Nets* Var. Elimination *Markov Processes* Value Iteration

Representation

Reasoning Technique

How can we minimally extend Markov Chains?



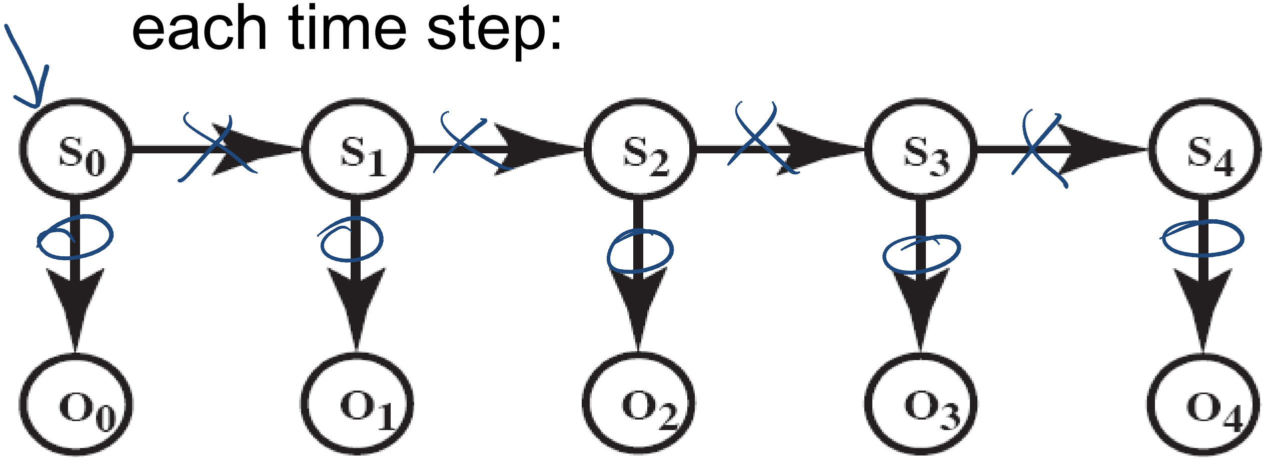
- Maintaining the Markov and stationary assumption

A useful situation to model is the one in which:

- the reasoning system **does not have access** to the states
- but can **make observations** that give some information about the current state

Hidden Markov Model

- A Hidden Markov Model (HMM) starts with a Markov chain, and adds a noisy observation about the state at each time step:

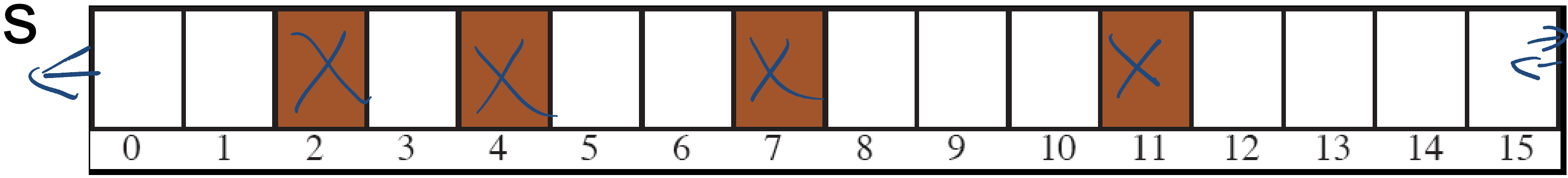


- |domain(S)| = k
- |domain(O)| = h

- $P(S_0)$ specifies initial conditions   K
- $P(S_{t+1}|S_t)$ specifies the dynamics   K × K
- $P(O_t|S_t)$ specifies the sensor model   K × h { K prob. dist. over O }

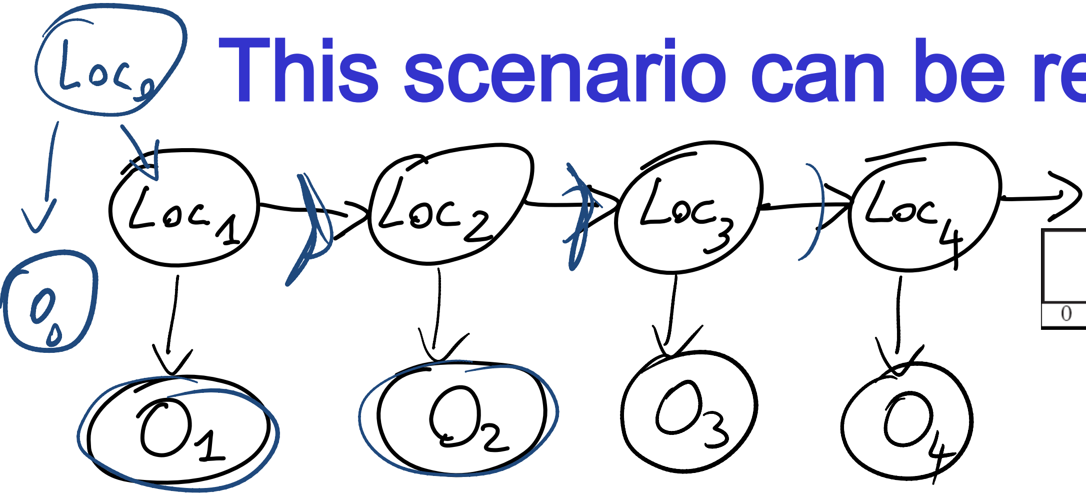
Example: Localization for "Pushed around" Robot

- **Localization** (where am I?) is a fundamental problem in robotics
- Suppose a robot is in a circular corridor with 16 locations



  - There are four doors at positions: 2, 4, 7, 11
  - The Robot initially doesn't know where it is
  - The Robot is pushed around. After a push it can stay in the same location, move left or right.
  - The Robot has Noisy sensor telling whether it is in front of a door

This scenario can be represented as…



- **Example Stochastic Dynamics**: when pushed, it stays in the same location p=0.2, moves left or right with equal probability

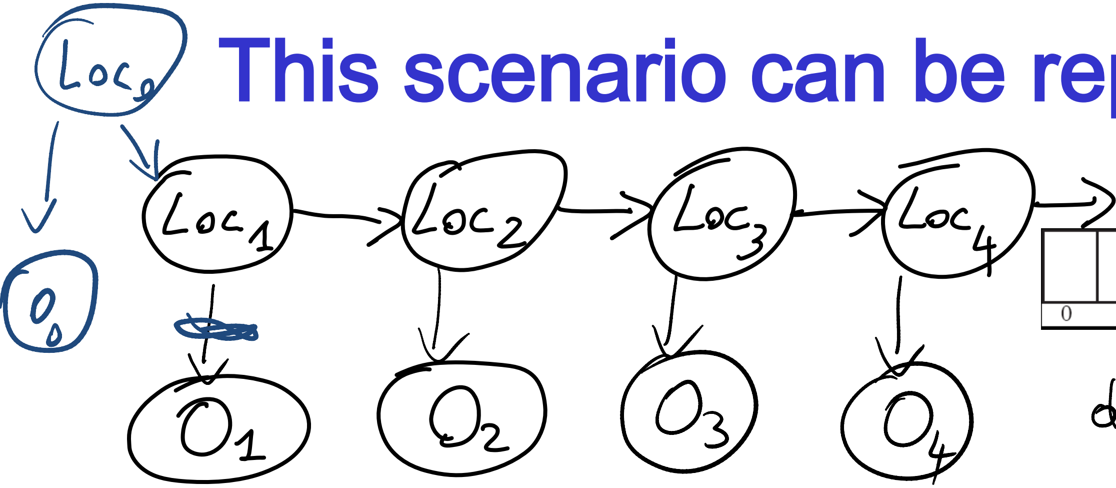
$P(Loc_{t+1} / Loc_t)$

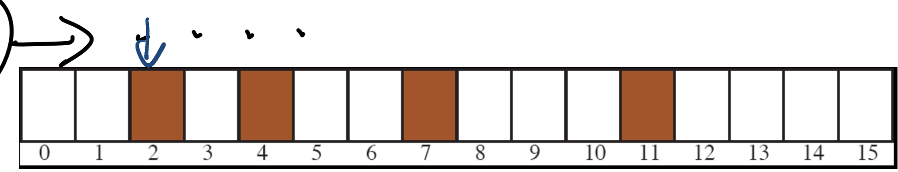
Loc_t \ Loc_{t+1}	0	1	2	....	15
0	.2	.4	0	... 0	.4
1	.4	.2	.4	0	0
2					
3					
⋮					
15					

$P(Loc_0) =$

0	1	2	…	15
1/16	1/16	1/16	…	

This scenario can be represented as…





$dom(Loc_i) = \{0, 1, \ldots, 15\}$

Example of Noisy sensor telling whether it is in front of a door.

- If it is in front of a door $P(O_t = T) = .8$
- If not in front of a door $P(O_t = T) = .1$

$\boxed{P(O_t \mid Loc_t)}$

Loc_t	$P(O_t=T)$	$P(O_t=F)$
0	.1	.9
1	.1	.9
2	.8	.2
3	.1	.9
4	.8	.2
⋮		

16 prob. distributions

Useful inference in HMMs

- **Localization**: Robot starts at an unknown location and it is pushed around *t* times. It wants to determine where it is

$$P(Loc_t \mid o_0, o_1, \ldots, o_t)$$

- **In general (Filtering):** compute the posterior distribution over the current state given all evidence to date

$$P(X_t \mid o_{0:t}) \quad or \quad P(X_t \mid e_{0:t})$$

Another very useful HMM Inference

- **Most Likely Sequence** (given the evidence seen so far)

$$\text{argmax}_{x_{0:t}} P(X_{0:t} \mid e_{0:t})$$

HMM: more agile terminology

Formal Specification as five-tuple (S, K, Π, A, B)

$S = \{s_1, \ldots, s_N\}$ Set of States

$K = \{o_1, \ldots, o_M\} = \{1, \ldots, M\}$ Output Alphabet

$\Pi = \{\pi_i\}, i \in S$ Initial State Probabilities

$A = \{a_{ij}\}, \quad i, j \in S \qquad \sum_{j=1}^{N} a_{ij} = 1$ State Transition Probabilities

$B = \{b_i(o_t)\}, \quad i \in S, \quad o_t \in K \qquad \sum_{t=1}^{M} b_i(o_t) = 1$ Symbol Emission Probabilities

The forward procedure

$$\boxed{\alpha_t(i) = P(o_1 o_2 \ldots o_t, X_t = i)}$$

compute this and then simply normalize

1. Initialization

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N$$

2. Induction

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i)\, a_{ij}\, b_j(o_t), \quad 1 \le i \le N, 1 \le j \le N$$

$$P(X_t = i \mid o_1 \ldots o_t) = \frac{\alpha_t(i)}{\sum_{J}^{N} \alpha_t(J)}$$

Complexity☺

$\approx$

evidence / observations

door ¬door door door

$$\alpha_1(i) = \pi_i b_i(o_1)$$

states

$\emptyset$ $\alpha_1(\emptyset)$ $\alpha_2(\emptyset)$

1 $\alpha_1(1)$

2

3

4

5

6

7

8

16

$\sim \frac{1}{16}$

$$\alpha_1(\emptyset) = \pi_\emptyset \, b_\emptyset(\text{door})$$

$$\alpha_2(\emptyset) = \sum_{i=1}^{N} \alpha_1(i)\, a_{i\emptyset}\, b_\emptyset(\neg\text{door})$$

generalize state

$$\alpha_2(j) = \sum_{i=1}^{N} \alpha_1(i)\, a_{ij}\, b_j(\neg\text{door})$$

generalize time slice

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

Finding the Best State Sequence :

$$\boxed{\arg\max_X P(X \mid O)}$$

$v_t(j)$: probability of the most probable path that leads to that node

The Viterbi Algorithm:

Initialization: $v_1(j) = \pi_j b_j(o_1)$, $1 \le j \le N$

Induction: $v_t(j) = \max_{1 \le i \le N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$, $1 \le j \le N$

Store backtrace:

$$bt_t(j) = \arg\max_{1 \le i \le N} v_{t-1}(i)\, a_{ij}\, \cancel{b_j(o_t)}, \quad 1 \le j \le N$$

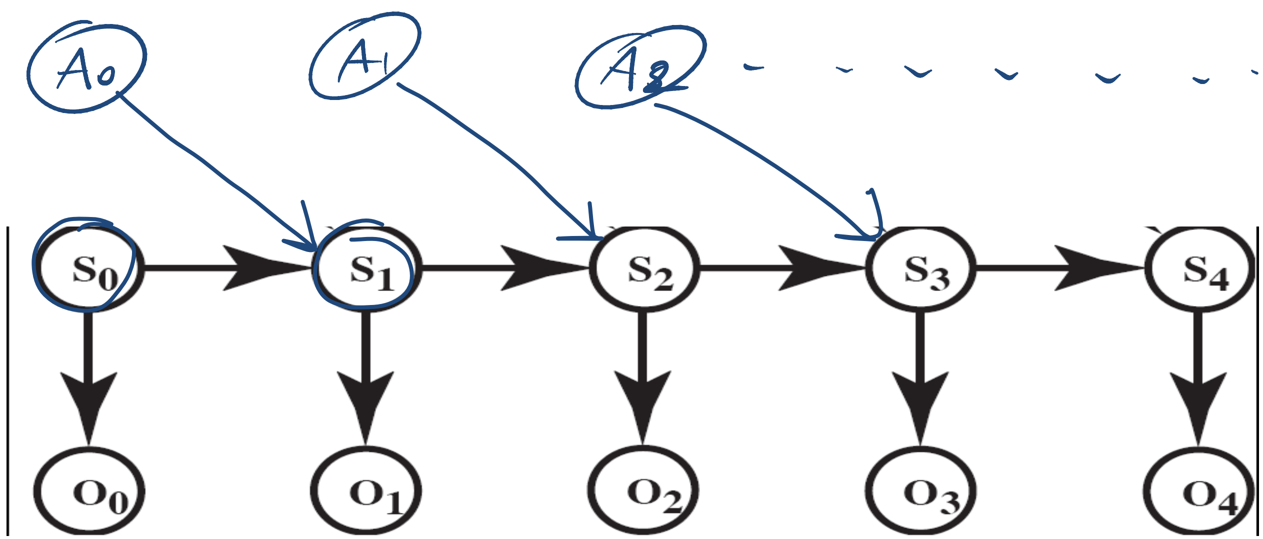
- Termination and path readout:

$P(X|O) = \max_{1 \le i \le N} v_t(i)$

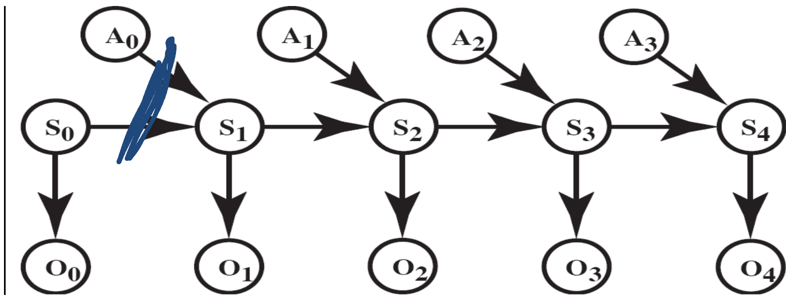
- To recover X start from $bt_t(i) = \arg\max_{1 \le i \le N} v_t(i)$ and keep going backward

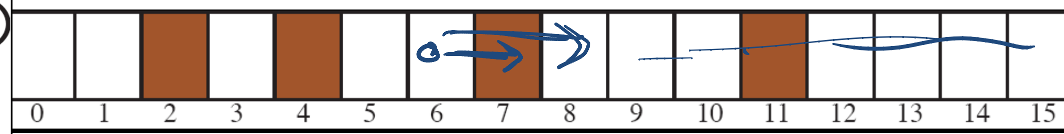
Robot Localization: More complex Example

- Suppose a robot wants to determine its location based on its actions and its sensor readings
- Three actions: *goRight, goLeft, Stay*
- This can be represented by an augmented HMM



Robot Localization Sensor and Dynamics Model





$P(O_t \mid Loc_t)$

- **Sample Sensor Model** (assume same as for pushed around)
- **Sample Stochastic Dynamics**: $P(Loc_{t+1} \mid Action_t, Loc_t)$

$P(Loc_{t+1} = L \mid Action_t = goRight, Loc_t = L) = 0.1$

$P(Loc_{t+1} = L+1 \mid Action_t = goRight, Loc_t = L) = 0.8$

$P(Loc_{t+1} = L + 2 \mid Action_t = goRight, Loc_t = L) = 0.074$

$P(Loc_{t+1} = L' \mid Action_t = goRight, Loc_t = L) = 0.002$ *for all other locations L'*

- All location arithmetic is modulo 16
- The action *goLeft* works the same but to the left

Dynamics Model More Details

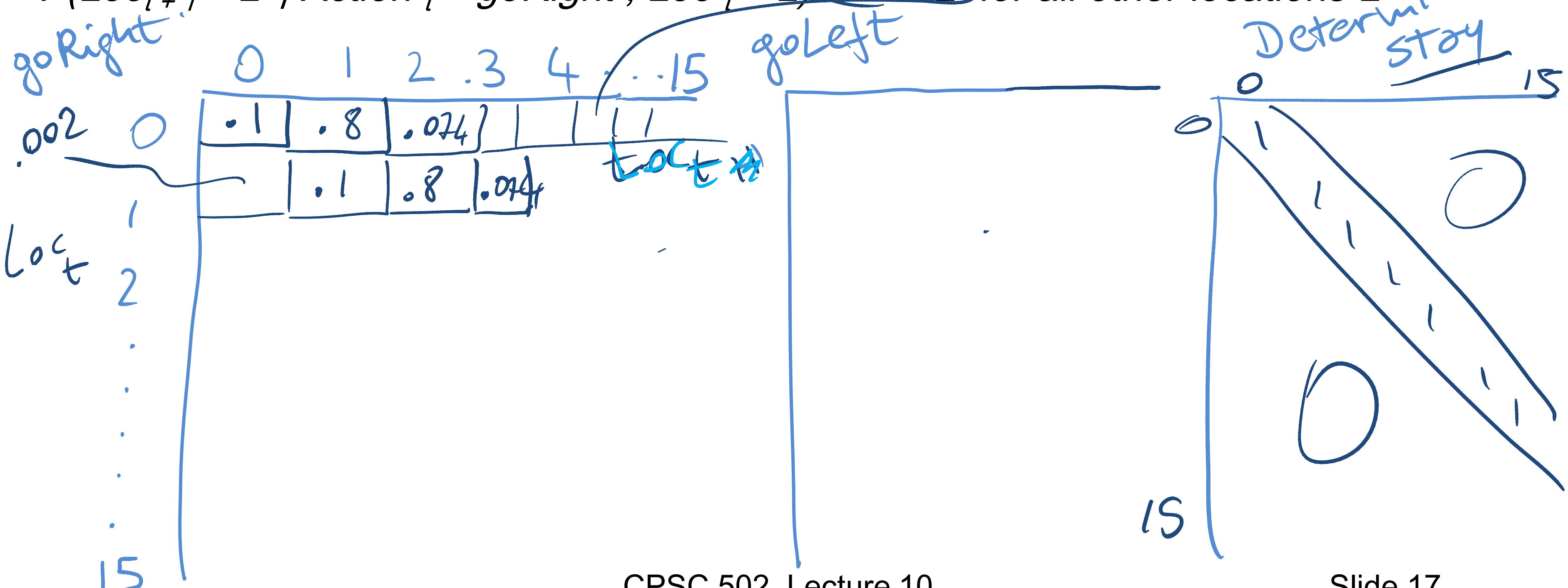
- **Sample Stochastic Dynamics:** $P(Loc_{t+1} \mid Action, Loc_t)$

$P(Loc_{t+1} = L \mid Action_t = goRight, Loc_t = L) = \boxed{0.1}$

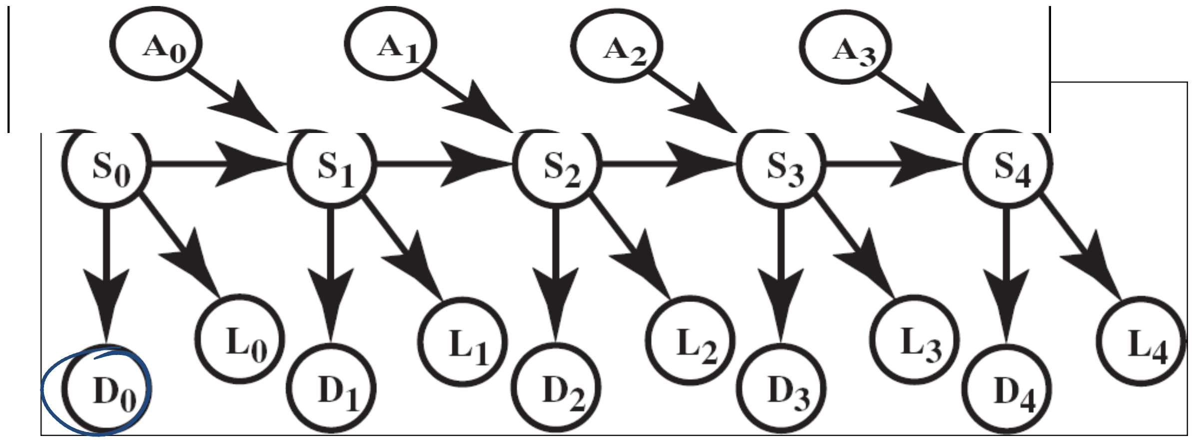
$P(Loc_{t+1} = L+1 \mid Action_t = goRight, Loc_t = L) = 0.8$

$P(Loc_{t+1} = L+2 \mid Action_t = goRight, Loc_t = L) = \underline{0.074}$

$P(Loc_{t+1} = L' \mid Action_t = goRight, Loc_t = L) = 0.002$ for all other locations L'



Robot Localization additional sensor



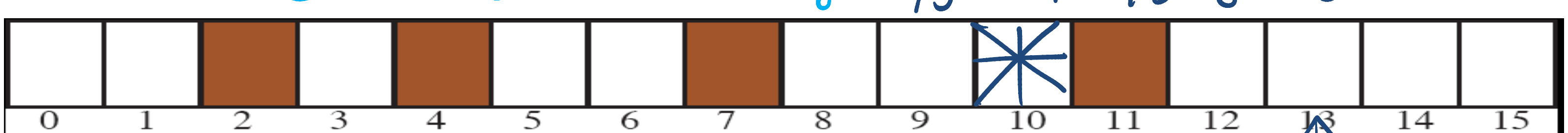
$L_t = T$

the Robot senses light

- **Additional Light Sensor:** there is light coming through an opening at location 10 $P(L_t \mid Loc_t)$

$P(L_t=F)$ .2 .05 .01 .05 .2 .4

$P(L_t=T)$ .8 .95 .99 .95 .8 .6



- **Info from the two sensors is combined :"Sensor Fusion"**

The Robot starts at an unknown location and must determine where it is

The model appears to be too ambiguous

- Sensors are too noisy
- Dynamics are too stochastic to infer anything

But inference actually works pretty well.

check:

`http://www.cs.ubc.ca/spider/poole/demos/localization/localization.html`

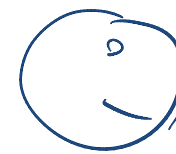
It uses a generalized form of filtering (not just a sequence of observations, but a sequence of observation pairs (from the two sensors) + the actions

HMMs have many other applications….

Natural Language Processing: e.g., Speech Recognition

- *States:* phoneme \ word
- *Observations*: acoustic signal \ phoneme

HMM 1 HMM 2



My name is

Bioinformatics: Gene Finding

- *States:* coding / non-coding region   x x ✓ ✓ ✓ x x
- *Observations:* DNA Sequences → A T C G G A A

For these problems the critical inference is:

find the most likely sequence of states given a sequence of observations

Viterbi Algo

TODO for next Tue

Start Reading Chp 9 of textbook (up tp 9.4 included)

Work on assignment 2

Sample scenario to explore in demo

- Keep making observations without moving. What happens?
- Then keep moving without making observations. What happens?
- Assume you are at a certain position alternate moves and observations
- ….

NEED to explain Filtering

Because it will be used in POMDPs

Markov Models

Markov Chains

Simplest Possible Dynamic Bnet

Hidden Markov Model

Add noisy Observations about the state at time t

POMDP

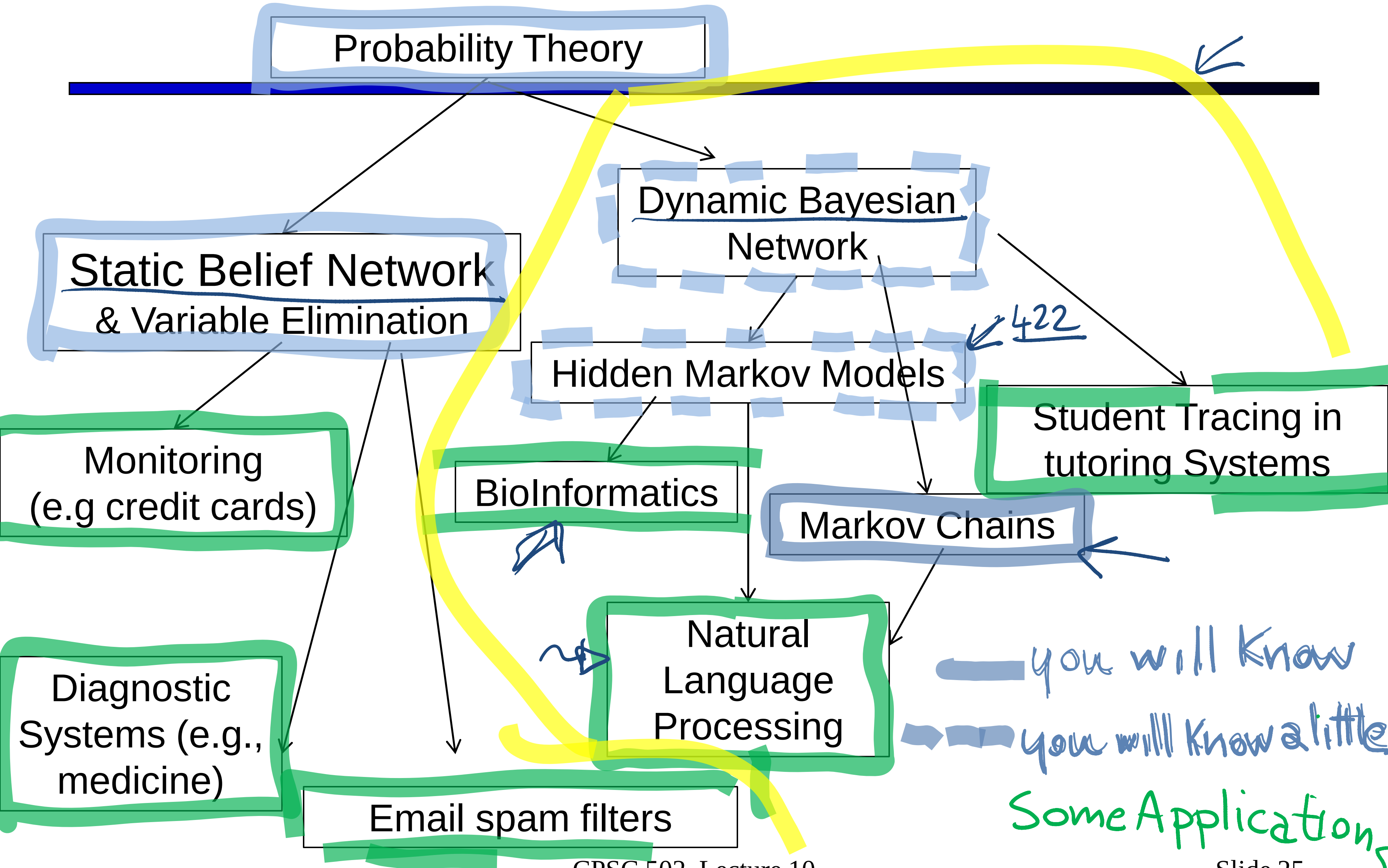
Partially observable

Add Actions and Values (Rewards)

Markov Decision Processes (MDPs)

in CPSC 4 2 2

Answering Query under Uncertainty



Lecture Overview

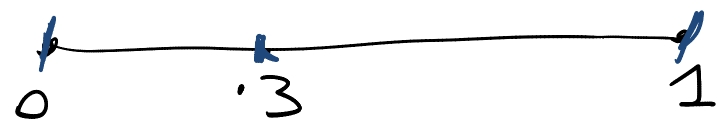
- **Recap**
- **Temporal Probabilistic Models**
- Start Markov Models
  - Markov Chain
  - Markov Chains in Natural Language Processing

Sampling a discrete probability distribution

e.g. Sim. Annealing. Select n' with probability P

P= .3

generate random number in [0,1]



If < .3 accept n'

e.g. Beam Search: Select K individuals. Probability of selection proportional to their value

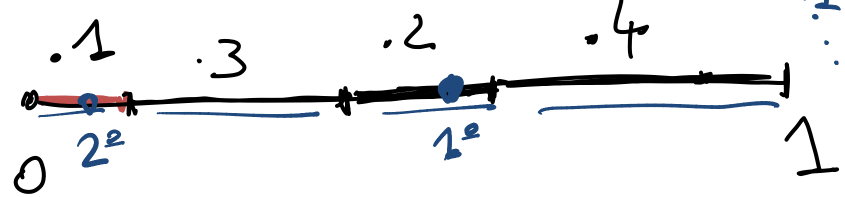
SAME HERE

→ n_1 $\boxed{P_1 = .1}$

→ n_2 $\boxed{P_2 = .3}$

→ n_3 $\boxed{P_3 = .2}$

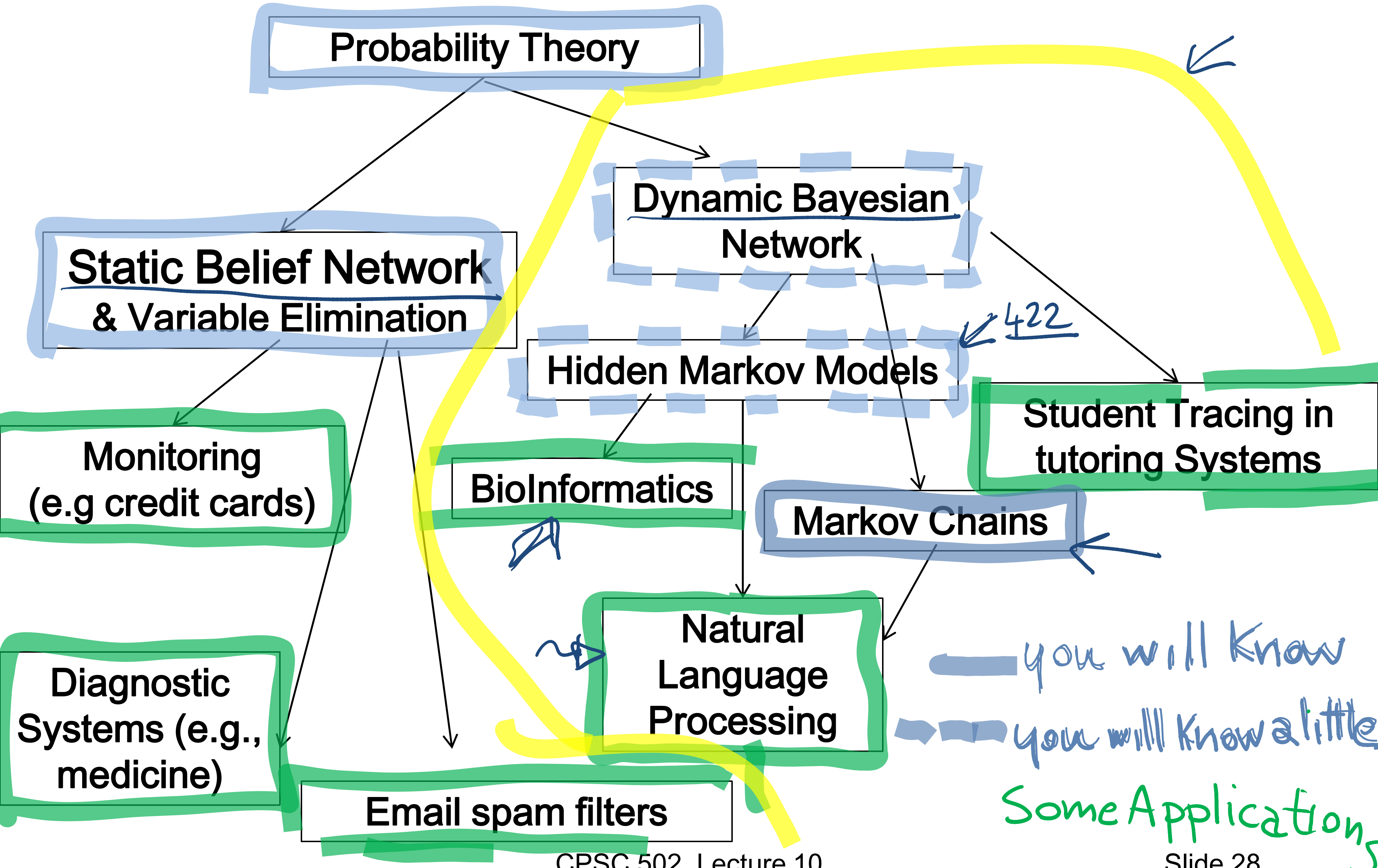
→ n_4 $\boxed{P_4 = .4}$



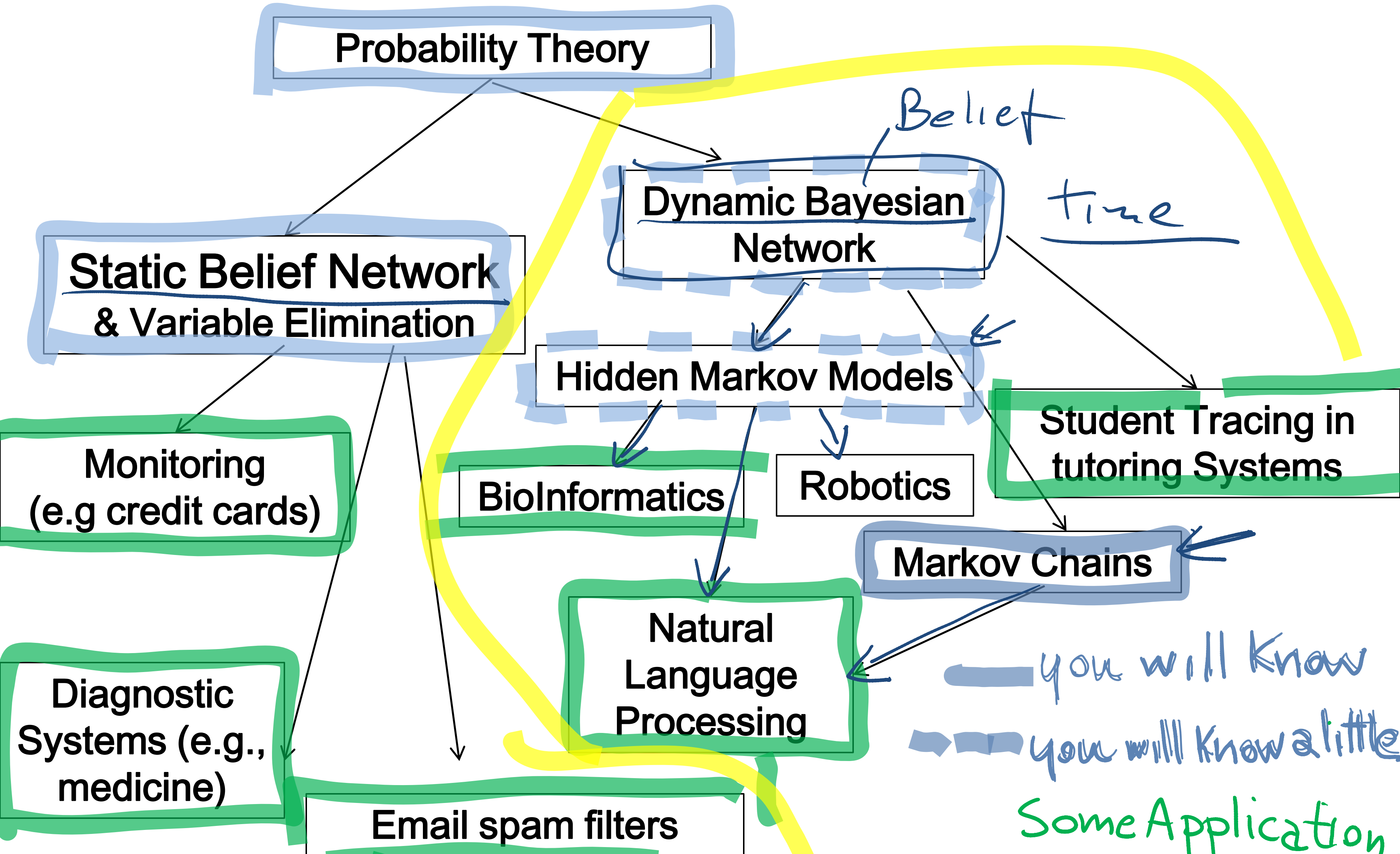
n_3 first sample

n_1 second sample

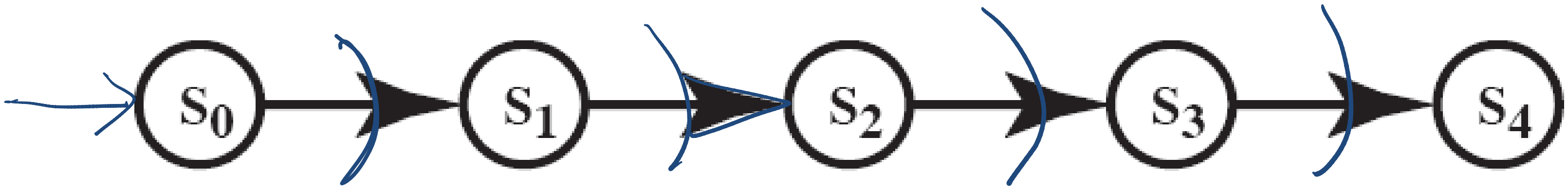
Answering Query under Uncertainty



Answering Queries under Uncertainty



Stationary Markov Chain (SMC)



$|dom(s_i)| = K$

A stationary Markov Chain : for all t >0

- $P(S_{t+1}| S_0,\ldots,S_t) = P(S_{t+1}|S_t)$ and
- $P(S_{t+1}| S_t)$ the same $\forall t$

We only need to specify $P(S_0)$ K and $P(S_{t+1}|S_t)$

$K \times K$

- Simple Model, easy to specify
- Often the natural model
- The network can extend indefinitely
- **Variations of SMC are at the core of most Natural Language Processing (NLP) applications!**

K prob distrib.